

Programmation (les Tableaux)

Exercice 1

Ecrire un programme en C qui lit la dimension N d'un tableau T de nombres réels (de dimension maximale 50), remplit le tableau par des notes entrées au clavier et affiche le tableau. Calculer et afficher ensuite la somme, le produit et la moyenne des éléments du tableau.

Exercice 2

Ecrire un programme en C qui fait la lecture d'un entier n, qui stocke sa conversion binaire dans un tableau T (tant que le nombre décimal est différent de 0, on le divise par deux et on met le reste dans le tableau.) et qui affiche vers la fin ce nombre en binaire.

Rappel : $r = m \% q$; (signifie : affecte à la variable r le reste de la division de m par q)

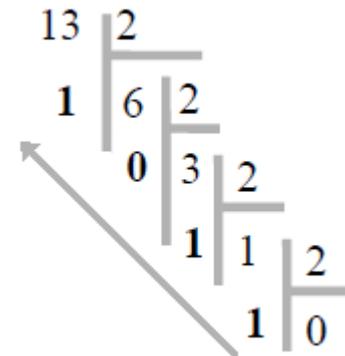
Exemple : n=13

Le programme doit calculer :

$A[0]=1, A[1]=0, A[2]=1, A[3]=1$

Et doit afficher :

$(13)_{10} = (1101)_2$



Exercices 3

On considère un tableau de taille $N_{max} = 100$. Ce tableau contient les notes des étudiants de votre groupe. Écrivez un programme en C où vous déclarez ce tableau, demandez un entier n qui représente le nombre des étudiants puis réalisez les manipulations suivantes :

1. Saisir les n notes dans le tableau puis les afficher ;
2. Calculer et afficher la moyenne des notes ;
3. Chercher et afficher la note minimale ;
4. Chercher et afficher la note maximale ;
5. Chercher une notes donnée s'elle existe dans le tableau ou pas ;
6. Calculer et afficher le nombre d'occurrence d'une note donnée ;
7. Modifier une note donnée du tableau T connaissant son indice, et afficher le tableau après modification.
8. Trier le tableau T dans l'ordre croissant (**tri par sélection**).

Exercice 4

Ecrire un programme en C qui permet de lire les dimensions (L, C) d'un tableau T à deux dimensions de type réel, de saisir les données du tableau, de calculer et d'afficher la somme de toutes ses composantes, de calculer et d'afficher la somme de chaque ligne. La somme de la ligne i est stockée à la case $T[i, C+1]$. Le programme doit afficher les éléments du tableau.

Exercice 5

Soit M un tableau à deux dimensions d'éléments de type int, de taille n lignes et m colonnes. Un élément $M[i][j]$ est dit « creux » s'il est inférieur strictement à tous ses voisins (voir l'exemple).

Ecrire un programme en C où vous réalisez les opérations suivantes :

1. Demander à l'utilisateur de donner les dimensions n et m.
2. Saisir les éléments du tableau M.
3. Afficher les éléments creux (valeurs et indices) du tableau M donné, ainsi que le nombre de ces points creux.
4. Calculer et afficher le maximum et le minimum des éléments du tableau M.

Exemple :

| | | | | |
|----|---|---|---|----|
| 1 | 3 | 4 | 0 | 3 |
| 3 | 4 | 5 | 2 | 1 |
| 10 | 2 | 4 | 9 | 7 |
| 5 | 7 | 5 | 1 | 6 |
| 6 | 4 | 1 | 3 | 10 |

L'élément 2 d'indices (2,1) est creux.

Exercice 6

L'objectif de cet exercice est d'écrire un programme en C qui permet de chercher et d'afficher les *points-Cols* d'une matrice. On appelle *point-Col*, un élément d'une matrice qui est à la fois un maximum sur sa ligne et un minimum sur sa colonne (voir exemple). Dans ce cas, on va travailler avec une matrice carrée M de taille n x n où n est une constante et on suppose que tous les éléments de la matrice sont différents.

Suivez les étapes ci-dessous :

1. Donner les déclarations des variables nécessaires,
2. Lire la taille de la matrice M,
3. Ecrire les instructions pour remplir la matrice M,
4. Ecrire les instructions pour afficher la matrice M,
5. Ecrire les instructions nécessaires pour trouver et afficher les points-cols.

Exemple :

Si la matrice M =

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

Le programme doit afficher : L'élément $M[0][2] = 3$ est un point-col.

TD N°2
Programmation (les Tableaux)

Exercice 1

Ecrire un programme en C qui lit la dimension N d'un tableau T de nombres réels (de dimension maximale 50), remplit le tableau par des notes entrées au clavier et affiche le tableau. Calculer et afficher ensuite la somme, le produit et la moyenne des éléments du tableau.

Solution

```
#include<stdio.h>
#include<conio.h>
main()
{
    int i,n;
    float T[100],S=0,P=1,M;
    printf("Donner le nombre d'étudiants \n");
    scanf("%d",&n);
    // boucle for pour remplir le tableau, calculer la somme et la moyenne des notes
    for(i=0;i<n;i++)
    {
        printf("Donner la note %d\t",i+1);
        scanf("%f",&T[i]);
    }
    // boucle for pour afficher les elements du tableau
    for(i=0;i<n;i++)
    {
        printf("%.2f\t",T[i]);
    }
    printf("\n");
    // boucle for pour calculer la somme, le produit et la moyenne des notes
    for(i=0;i<n;i++)
    {
        S=S+T[i];
        P=P*T[i];
    }
    M=S/n;
    printf ("la somme des %d notes = %.2f\n",n,S);
    printf ("le produit des %d notes = %.2f\n",n,P);
    printf("la moyenne des %d notes = %.2f\n",n,M);
}
```

Exercice 2

Ecrire un programme en C qui fait la lecture d'un entier n, qui stocke sa conversion binaire dans un tableau T (tant que le nombre décimal est différent de 0, on le divise par deux et on met le reste dans le tableau.) et qui affiche vers la fin ce nombre en binaire.

Rappel : $r = m \% q$; (signifie : affecte à la variable r le reste de la division de m par q)

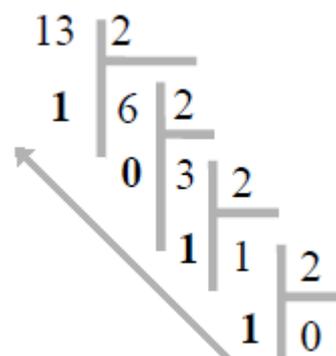
Exemple : n=13

Le programme doit calculer :

A[0]=1, A[1]=0, A[2]=1, A[3]=1

Et doit afficher :

$(13)_{10} = (1101)_2$



Solution

```
#include<stdio.h>
#include<conio.h>
main()
{
    int n, i=0, j ;
    int T[50];
    printf("donner un entier \n");
    scanf("%d",&n);
    printf("(%d)10 = ",n);
    do
    {
        T[i]=n%2;
        n=n/2;
        i=i+1;
    }
    while (n!=0) ;
    for(j=i-1;j>=0;j--)
        printf("%d\t",T[j]);
    getch();
}
```

Exercices 3

On considère un tableau de taille $N_{max} = 100$. Ce tableau contient les notes des étudiants de votre groupe. Écrivez un programme en C où vous déclarez ce tableau, demandez un entier n qui représente le nombre des étudiants puis réalisez les manipulations suivantes :

9. Saisir les n notes dans le tableau puis les afficher;
10. Calculer et afficher la moyenne des notes ;
11. Chercher et afficher la note minimale ;
12. Chercher et afficher la note maximale ;
13. Chercher une notes donnée s'elle existe dans le tableau ou pas;
14. Calculer et afficher le nombre d'occurrence d'une note donnée ;
15. Modifier une note donnée du tableau T connaissant son indice i et afficher le tableau après modification.
16. Insérer une note à un indice i donné et afficher le tableau après insertion.
17. Trier le tableau T dans l'ordre croissant (tri par sélection).

Solution

```
#include<stdio.h>
#include<conio.h>
main()
{
    int i,n,k,a,j, c;
    float T[100],tempo,S=0,M,max,min,X,x;
    /* i,j,c = compteur
    n = nombre d'étudiant
    S = sommes des notes
    M = Moyennes des notes
    max = maximum, min = minimum
    k = l'indice du max et du min
    X = élément à chercher dans le tableau*/

    printf("Donner le nombre d'étudiants \n");
```

```

scanf("%d",&n);

// boucle for pour remplir le tableau, calculer la somme et la moyenne des notes
for(i=0;i<n;i++)
{
    printf("Donner la note %d\t",i+1);
    scanf("%f",&T[i]);
    S=S+T[i];
}
    M=S/n;
for(i=0;i<n;i++)
printf("%f\t",T[i]);
printf("\nla somme des %d notes = %.2f\n",n,S);
printf("la moyenne des %d notes = %.2f\n",n,M);

/* boucle for pour trouver et afficher la valeur de la note maximale */
max=T[0];
k=0;
for (i=1;i<n;i++)
    if(T[i]>max)
    {
        max=T[i];
        k=i;
    }
printf("la note maximale est %.2f et se trouve a la position
        %d\n", max, k);
/* boucle for pour trouver et afficher la valeur et l'indice de la
note minimale */
min=T[0];
k=0;
for (i=1;i<n;i++)
    if(T[i]<min)
    {
        min=T[i];
        k=i;
    }
printf("la note minimale est %.2f et se trouve a la position %d\n", min,k);

/* boucle for pour chercher et afficher une note donnée
printf("donner la note que vous voulez chercher dans le tableau \n");
scanf("%f",&X);
a=0;
for (i=0;i<n;i++)
    if(X==T[i])
        a=1;
    if(a==1)
        printf("la note %.2f se trouve dans le tableau \n",X);
    else
        printf("la note %.2f ne se trouve pas dans le tableau\n",X);
*/
//boucle while pour chercher et afficher une note donnée
printf("donner la note que vous voulez chercher dans le tableau \n");
scanf("%f",&X);

```

```

    i=0;
    a=0;
while((i<n) && (a==0))
{
    if(T[i]==X)
        a=1;
    i++;
}
if(a==1)
    printf("la note %.2f se trouve dans le tableau\n",X);
else
    printf("la note %.2f ne se trouve pas dans le tableau\n",X);
/* boucle for pour Calculer et afficher le nombre d'occurrence d'une note donnée dans le tableau */
printf("donner la note pour laquelle vous voulez chercher le nombre d'occurrence \n");
    scanf("%f",&X);
    c=0;
    for (i=0;i<n;i++)
        if (T[i] == X)
            c+=1;
        printf("le nombre d'occurrence de la note %.2f = %d\n",X,c);
/* Pour modifier une note donnée a un indice i donné*/
printf(" \n entrer l'indice du note à modifier et la nouvelle note \t");
scanf("%d %f",&i,&x);
T[i]=x;
//affichage du tableau apres la modification
for(i=0;i<n;i++)
    printf("%.2f \t",T[i]);
    printf("\n");
/* Pour insérer une note a un indice i donné*/
printf("\n Donner la note a insérer \t");
scanf("%f",&X);
printf("Donner l'indice ou vous voulez l'insérer\t");
scanf("%d",&k);
    for(i=n;i>k;i--)
        T[i]=T[i-1];
        T[k]=X;
        n=n+1;

/*boucle for pour afficher les éléments du tableau après avoir inséré une note */
printf("\n\nle tableau après avoir inséré la note \n");
for(i=0;i<n;i++)
    printf ("%.2f\t",T[i]);

/* trie du tableau*/
for(i=0;i<(n-1);i++)
    for(j=(i+1);j<n;j++)
        if(T[i]>T[j])
            {
                tempo=T[i];
                T[i]=T[j];
                T[j]=tempo;
            }

```

```

/* boucle for pour afficher les éléments du tableau après l'avoir trié*/
printf("\n\n Le tableau après l'avoir trié \n");
for(i=0;i<n;i++)
    printf ("%f",T[i]);
getch();
}

```

Exercice 4

Ecrire un programme en C qui permet de lire les dimensions (**L**, **C**) d'un tableau **T** à deux dimensions de type réel, de saisir les données du tableau, de calculer et d'afficher la somme de toutes ses composantes, de calculer et d'afficher la somme de chaque ligne. La somme de la ligne **i** est stockée à la case **T [i, C+1]**. Le programme doit afficher les éléments du tableau.

Solution

```

#include<stdio.h>
#include<conio.h>
main()
{
    float T[50][50],S;
    int i,j,n,m;

//lecture de la taille du tableau
    do {
        printf("entrer les dimensions du tableau \n");
        scanf("%d%d",&n,&m);
    }
    while(n<=0 || n>50 || m<=0 || m>50);
//Saisis des éléments du tableau
    for(i=0;i<n;i++)
    {
        for(j=0;j<m;j++)
        {
            printf("entrer T[%d,%d]= \t",i,j) ;
            scanf("%f",&T[i][j]);
        }
    }
//calcul de la somme de tous les éléments du tableau
    for(i=0;i<n;i++)
    {
        for(j=0;j<m;j++)
            S=S+T[i][j];
    }
    printf("\n La somme est S= %.2f\n",S);

//calcul de la somme de chaque ligne
    for(i=0;i<n;i++)
    {
        S=0;
        for(j=0;j<m;j++)
            S=S+T[i][j];
        T[i][m]=S;
    }
}

```

```
//affichage de la somme de chaque ligne
for(i=0;i<n;i++)
{
for(j=0;j<=m;j++)
printf("%.2f\t",T[i][j]);
printf("\n");
}
getch();
}
```

Exercice 5

Soit M un tableau à deux dimensions d'éléments de type int, de taille n lignes et m colonnes. Un élément M[i][j] est dit « creux » s'il est inférieur strictement à tous ses voisins (voir l'exemple).

Ecrire un programme en C où vous réalisez les opérations suivantes:

1. Demander à l'utilisateur de donner les dimensions n et m.
2. Saisir les éléments du tableau M.
3. Afficher les éléments creux (valeurs et indices) du tableau M donné, ainsi que le nombre de ces points creux.
4. Calculer et afficher le maximum et le minimum des éléments du tableau M.

Exemple :

| | | | | |
|----|---|---|---|----|
| 1 | 3 | 4 | 0 | 3 |
| 3 | 4 | 5 | 2 | 1 |
| 10 | 2 | 4 | 9 | 7 |
| 5 | 7 | 5 | 1 | 6 |
| 6 | 4 | 1 | 3 | 10 |

L'élément 2 d'indices (2,1) est creux.

Solution

```
#include<conio.h>
#include<stdio.h>
main()
{
int k,l,a=0,cont=0,i,j,n,m,T[50][60],max,min,p;
printf("donner n et m\n");
scanf("%d%d",&n,&m);
/*saisir la matrice T*/
for(i=0;i<n;i++)
for(j=0;j<m;j++)
{
printf("T[%d][%d]=",i,j);
scanf("%d",&T[i][j]);
}
/*Affichage de la matrice*/
for(i=0;i<n;i++)
{
for(j=0;j<m;j++)
printf("%d\t",T[i][j]);
printf("\n");
}

/* Affichage des points creux et leurs indices */
```

```

/* Calcule et affichage du nombres des points creux */
for(i=1;i<n-1;i++)
{
  for(j=1;j<m-1;j++)
  {
    for(k=i-1;k<=i+1;k++)
    {
      for(l=j-1;l<=j+1;l++)
      {
        if(T[i][j]<T[k][l])
          a++;
        if (a==8)
          {
            printf("\nL'element %d d'indice (%d,%d) est un element creux\n",T[i][j],i,j);
            cont++;
          }
      }
    }
  }
}
printf("\nle nombre des element creux est %d\n",cont);

```

```

/*Calcule du Max et du Min*/
max=T[0][0];
min=T[0][0];
for(i=0;i<n;i++)
  for(j=0;j<m;j++)
    {
      if (max<T[i][j]) max=T[i][j];
      if (min>T[i][j]) min=T[i][j];
    }
printf("\nMin=%d et Max=%d\n",min,max);
getch();
}

```

Exercice 6

L'objectif de cet exercice est d'écrire un programme en C qui permet de chercher et d'afficher les *points-Cols* d'une matrice. On appelle *point-Col*, un élément d'une matrice qui est à la fois un maximum sur sa ligne et un minimum sur sa colonne (voir exemple). Dans ce cas, on va travailler avec une matrice carrée M de taille n x n où n est une constante et on suppose que tous les éléments de la matrice sont différents.

Suivez les étapes ci-dessous:

6. Donner les déclarations des variables nécessaires,
7. Lire la taille de la matrice M,
8. Ecrire les instructions pour remplir la matrice M,
9. Ecrire les instructions pour afficher la matrice M,
10. Ecrire les instructions nécessaires pour trouver et afficher les points-cols.

Exemple :

Si la matrice M =

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

Le programme doit afficher : L'élément $M[0][2] = 3$ est un point-col.

Solution

```
#include<stdio.h>
#include<conio.h>
int main()
{
  int i,j,n,k;
  int T[20][20],M[20], max,min;
  printf("donner la valeur de n : ");
  scanf("%d",&n);
  printf(" La saisi des élément du tableau T: \n");
  for (i=0;i<n;i++)
  {
    for (j=0;j<n;j++)
    {
      printf(" element N: %d %d : ",i,j);
      scanf("%d",&T[i][j]);
    }
  }
  printf(" Affichage des éléments du tableau T: \n ");
  for (i=0;i<n;i++)
  {
    for (j=0;j<n;j++)
    {
      printf("%5d",T[i][j]);
    }
    printf("\n");
  }
  //debut de la recherche du point-col

  //recherche du maximum sur les lignes
  k=0;
  for(i=0;i<n;i++)
  {
    max=T[0][0];
    for (j=0;j<n;j++)
    {
      if(T[i][j]>max)
      {
        max=T[i][j];
        k=j;
      }
    }
  }
  printf("\nle maximum sur la ligne %d =%d\n",i,max);
  //recherche du minimum sur les colonnes
  min= T[0][k];
  for (j=0;j<n;j++)
  if(T[j][k]<min)
  {
    min=T[j][k];
  }
  printf("\nle minimum sur la colone %d =%d\n",k,min);
  //comparaison du max sur la ligne et min sur la colone
```

```
if(max==min)
    printf("T[%d][%d]=%d est un point col",i,k,T[i][k]);
}
```